AARef: Exploiting Authorship Identifiers of Micro-Messages with Refinement Blocks

Sarp Aykent
Department of Computer Science
& Software Engineering
Auburn University
Auburn, AL, USA
sarp@auburn.edu

Gerry Dozier
Department of Computer Science
& Software Engineering
Auburn University
Auburn, AL, USA
doziegv@auburn.edu

***Abstract*—With the rise of web-based social networking, a great many short texts/micro-messages are exchanged daily. Although short texts/micro-messages are a powerful and efficient way to communicate among individuals, their anonymity and short-length attributes give rise to a real challenge for Author Identification studies. In this paper, we tackle the Author Identification of short texts problem via Convolutional Neural Networks (CNNs). Specifically, we present a novel Multi-Channel CNN architecture that processes different features of text via word, character, and parts of speech (POS) embeddings. We examine the usefulness of different feature types and show that the combination of embeddings can capture different stylometric features. In addition, we add an identity mapping block in the convolutional layer to preserve the maximum amount of information from features. Extensive experiments with a variety number of authors and writing samples per author were conducted using our proposed architecture. Based on the experiments, our proposed method outperforms the state-of-the-art system on a large Twitter dataset.**

***Index Terms*—Convolutional Neural Networks, Authorship Attribution, Author Identification, Identity Mapping**

I. Introduction

Author Identification [1], also referred to as Authorship Attribution, is the task of capturing the stylistic information in a collection of writing samples and identifying the authors of unknown texts based on the captured information [2]. Author Identification research has developed substantially over the last decade with a focus on long texts [3], [4]. Recently, there is a growing interest in identifying authors of micro-messages due to the development of social media platforms and the emergence of social media as the primary mode of communication [3]. The increase in micro-message traffic has attracted attention in many fields such as email author identification [5], blog author identification [6], and web message author identification [7].

The task of identifying authors of micro-messages has been shown to be more difficult than Author Identification using long texts [8]. Social media platforms, such as Twitter [9], limit the number of characters for each tweet (micro-message). Furthermore, many tweets are not as formal as book chapters or documents, may lack the syntax of natural languages and contexts and may contain a large number of typos and syntax errors. In this paper, we attempt to tackle these problems via a novel Multi-Channel CNN architecture.

The stylometric properties of a text can be captured by different feature sets [3]. Therefore we combine word embeddings, character embeddings, and part of speech embeddings within one CNN architecture to form a Multi-Channel CNN architecture, namely AARef. These embeddings can be considered as individual information channels for a CNN. We also added a novel multi branch convolution block with identity mapping in the convolutional layer. The identity mapping links the layers in the network and provides an alternative path to preserve the gradient. This skip connection ensures that the initial information can be captured and passed to the next layer. Our preliminary results of AARef using shared convolutional filters and skip connections for all embeddings outperforms the state-of-the-art methods [10]. We compare AARef with the best performing state-of-art methods within the literature [10] as well as several popular authorship attribution methods for micro-messages [4], [9].

We have designed a number of experiments with a varying number of authors (from 100 to 1,000) as well as a varying number of writing samples (from 50 to 500) per author. Each tweet was treated as one writing sample in the experiments. We show that one tweet can be successfully identified using our proposed Multi-Channel CNN architecture. Comparing with the state-of-art baselines, our model achieved a minimum of 3.81% improvement in classification accuracy on the Twitter dataset.

The remainder of this paper is organized as follows. Section II presents some work related to this study, Section III introduces our Multi-Channel CNN architecture for Author Identification in detail. Section IV and V present our experiments and results respectively. In Section VI, we summarize our work and briefly discuss the potential future research venues.

II. Related Work

In this section, we present earlier work related to our research.

A. *Author Identification*

In the literature [2], authors of the text samples are identified using a variety of methods. Memory-based learning was used by Luyckx et al. [11] to analyze the relationship between the number of authors, the number of writing samples per author, and the size of each writing sample. Luyckx et al. [11] showed that increasing the number of authors within a dataset has an adverse effect on Author Identification accuracy while increasing the number of writing samples per author as well as the size of each writing instance increases the identification rate.

Similarity-based [12]–[16] and statistical-based [12]–[16] methods are popular Author Identification techniques. Similarity-based methods calculate the distance between writing samples using the concept of relative [16] and cross-entropy [14]. Statistical-based methods use different sets of extracted features from writing samples to capture an author's writing style. Some of the successful methods include Support Vector Machines (SVMs) [17], Decision Trees [18], Random Forest Classifiers, k-Nearest Neighbors, Multilayer Perceptrons [19], Radial Basis Function Networks, and Naive Bayesian Classifiers, just name a few. There are comprehensive studies that investigated the performance of different feature sets [3], [12], [20]–[23]. Some of these feature sets include character and word n-grams [3], [12], [22], bag-of-words [20], part of speech(POS) [3], stylometry [23], and etc. A more comprehensive overview of the Author Identification analysis can be found in [2].

B. *Author Identification of Micro-Messages*

Schwartz et al. [9] introduced a method known as K-Signatures to identify authors of micro-messages. The K-Signatures method captures the style of an author by collecting the features only found in the writing samples of the author. Also, another condition for collecting the feature is that a given feature needs to be seen in k% of the author's writing samples. The K-Signature method was used by SVMs to classify an author of a given text.

Rocha et al. [3] provided an overview of the Authorship Identification methods used in social media. In addition, they provided performance analysis of the feature sets of micro-messages using two classifiers, namely, a Power Mean SVM (PMSVM) [24] and Random Forests. Rocha et al. [3] suggested that character 4-grams perform best among the features compared in the work, including word n-grams and Parts of Speech n-grams.

Shrestha et al. [25] used a different approach for identifying the authors of micro-messages via CNNs. The authors used a sequence of character n-grams as input. Similarly, the CNN architecture proposed by Theóphilo et al. [4] processes each micro-message as a sequence of character level 4-grams. While the architecture proposed by Theóphilo et al. [4] was not the best performer, it required less computational resources. Aykent et al. [10] proposed an architecture to efficiently use both word and character level n-grams as input. To the best of our knowledge, this is the best performing algorithm, to date, for Author Identification of micro-messages.

C. *Neural Networks for Author Identification*

CNNs for Author Identification have shown promising results [3], [10], [25], [26]. Kim [26] proposed a CNN for sentence classification. The CNN utilized convolutional filter sizes of 3, 4, and 5, a dropout rate, and max-over-time pooling (Collobert et al. [27]). Kim's approach [26] used a different combination of models such as: a) randomly initialized, b) static model, c) dynamic model, d) Multi-Channel that combines the static and dynamic models. The models, excluding the random model, used pre-trained word2vec word embeddings [28].

Recurrent Neural Networks (RNNs) [29], have also been used for Author Identification. Bagnall [29] used a RNN that predicts the next character in the sequence based on the previously seen characters. Different sets of probabilities of the next character for each author were generated, then authors were identified based on these probabilities. This was the best performing method for the PAN 2015 multi-language author identification competition.

D. *Convolutional Neural Networks*

Multi-Channel CNNs [30] have been extensively studied in some areas such as image classification, object detection, and speech recognization [31]. The color channels (such as RGB) in computer vision [30] or the wavelengths channels in speech recognization [31] have proven successful as Multi-Channel input for classification problems in their respective fields. Although natural language inputs are normally in the form of single-channel tokens or characters, the different sets of extracted features are shown to capture different stylometric features [3], [4], [12], [20], [22]. To our knowledge, no work has been done with respect to the micro-message author identification task using feature learning and model training from both word embeddings, character n-grams embeddings, and POS embeddings. In addition to the proposed novel Multi-Channel CNN architecture using three different embedding methods, we also proposed an identity mapping block added to the convolutional layer to allow for more information flow within the network. This identity mapping block greatly improves the performance of our model by preserving the information from the features [32].

III. Method

In this section, we present the details of AARef.

A. *Model Construction*

In light of the above discussion, we devised AARef architecture utilizing word embeddings, character n-gram embeddings, and POS embeddings. The Multi-Channel convolutional network architecture is shown in Figure 1, and the model details are as follows.

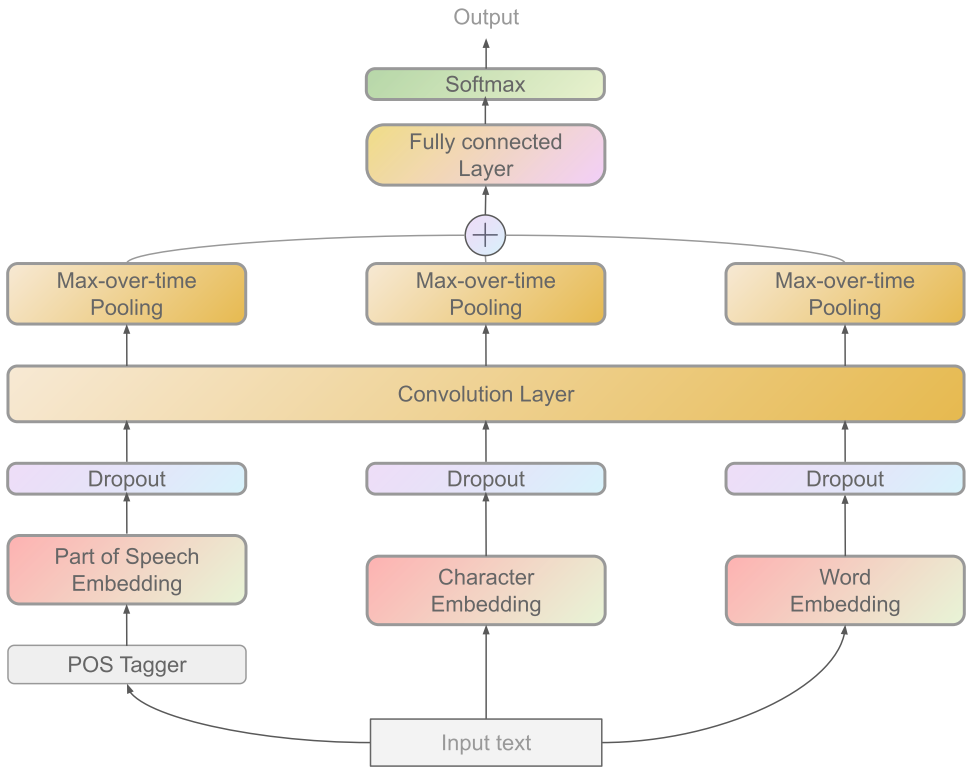

Fig. 1. AARef: Multi-Channel CNN architecture diagram. The writing samples are tokenized and fed to embeddings. Word embeddings, character bigram embeddings and POS embeddings are illustrated in the figure. The embeddings then forwarded to convolutional layers with the different window sizes. After convolution, max-over-time pooling is used in pooling layer, followed by merge layer, which combines the features by Add$(+)$ operation, then feed into a fully connected layer with softmax output.

1) Embeddings Layer: There are three embeddings used in parallel in our proposed network. For the word and character embeddings, the input text samples are tokenized and fed into each embedding separately. For the POS embedding, the input text is first tagged via TweetNLP Tagger [33]. After writing samples are converted into POS, they are tokenized and fed into POS embedding. All three embeddings are used as separate channels as shown in Figure 1. These feature embeddings are padded when necessary to have a constant size. Dropout is applied to the embeddings to avoid over-fitting. After dropout, the embeddings (inputs) are given to the convolutional layers with different window sizes in parallel.

2) Convolutional Layer: The three embedding channels are then separately forwarded to convolutional layers with different window sizes. Figure 2 shows a detailed illustration of the multi branch convolutional refinement blocks. The convolutional blocks apply different sized filters to the input, feature embeddings in our case, and shifts the filter until the end of the sequence. The window size of the convolution operation corresponds to the size of the filter that is applied to the input each time. Regarding the filters, two sets of filters are used with window sizes 3 and 4 separately.

For the convolutional branch with filter sizes 3 and 4, we used the Scaled Exponential Linear Unit (SELU) activation function [34] because of its self normalizing properties. The convolutional layer can be represented as follows:

$$f(x) = \lambda \begin{cases} x, & \text{if } x > 0 \\ \alpha e^{x} - \alpha, & \text{if } x <= 0 \end{cases} \tag{1}$$

where x is the output of the convolutional filters, and λ and α are pre-defined constants.

In addition to the convolution with window size 3 and 4, we add an identity mapping block on the left marked in a light grey color block. The features were upscaled to 500 first, then followed by two $L \times 3$ convolution operation with a batch normalization operation. An identity connection was added to the two convolution operation as shown in Figure 2 marked by a grey box. Given the output of the upscale layer, L, denoted as X_L, and the output of the identity block is X_{L+1}, the identity mapping block is represented by the following equation:

$$X_{L+1} = f(X_L) + X_L \tag{2}$$

This identity connection ensures that the information from the previous layer is not altered and can flow unimpeded to the next layer in the network.

For the convolutional layer identity mapping branch, we used a Leaky ReLU activation function [35]. The convolutional layer can be represented as follows:

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ 0.01x, & \text{if } x <= 0 \end{cases} \tag{3}$$

where x is the output of the convolutional filters.

Since we used character embedding, word embedding, and part of speech embedding as individual input channels, each channel has an output, feature map, of size $1,536(= 512 \times 3)$.

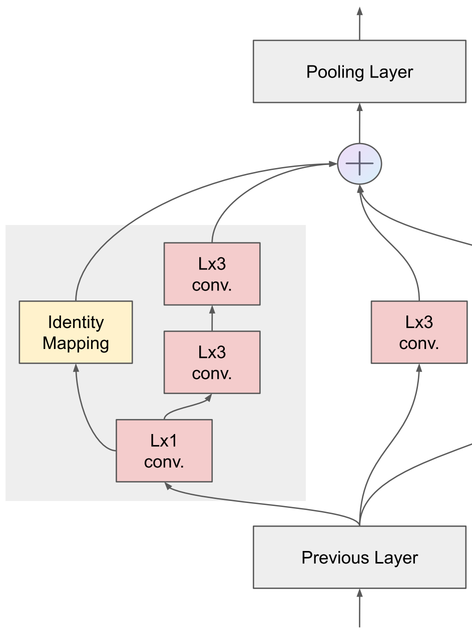

Fig. 2. The detailed illustration of the multi branch convolution blocks.

3) Pooling Layer: Convolutional layers are followed by max-over-time pooling [27], which takes the feature with maximum value in a given filter. For the feature matrix output x by the convolutional layer, the max-over-time pooling operation can output $\hat{x}$ as follows:

$$\left[\hat{x}\right]_i = \max_t \left[x\right]_{i,t} \quad 1 <= i <= M \tag{4}$$

where t is the position in the sentence, i is the convolutional filter applied and M is the numbers of filters for each window sizes as shown in Figure 1.

In this way, we make the network keep the most salient features produced by convolutional layers. Thus, the number of selected values with max-over-time pooling is equal to the number of filters in the architecture per feature embedding.

4) Merge Layer: The resulting features are merged into a single feature map using add (+) operations in merge layer as as shown in Figure 1. The add operation use pairwise add operation between feature vectors. The operation can be represented as follows:

$$x_{1..n} = a_{1..n} + b_{1..n} + c_{1..n} \tag{5}$$

where $x_{1..n}$ is the output of the merge layer. $a_{1..n}$, $b_{1..n}$, and $c_{1..n}$ represent the output of the last layer.

5) Fully Connected Layer with Softmax Output: After the merge layer, the final step is to feed the feature maps into the fully connected layer, followed by the softmax function for the classification. The softmax function, which is shown in equation 6, take exponents of each element x_i from the input feature map x and normalize them by dividing by the sum of all the exponentials. The equation is as follows:

$$S(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{k} e^{x_j}} \tag{6}$$

where k is the total number of inputs. Softmax function scores the authors based on the closeness of the feature map to the class of an author. The network guesses the author of the writing samples with the highest score.

B. Hyper-parameters

Hyper-parameters were selected using grid search [36]. The dimension of the embeddings used was $d = 400$. The dropout was 25%. For the convolutional layer, there were 512 filters per window size, $M = 512$, which makes a total of 1,536 filters. Hence, only one feature was selected per filter. We use a batch size of 64 for the experiment. The epoch limit was 100 with an early stopping condition. Early stopping was based on validation accuracy improvement and stops after 20 epochs without an improvement. The learning rate updated during the training by Adam optimizer. The initial learning rate was $1e-4$.

IV. Experiments

In this section, we present our experiments and datasets in detail.

The parameters used in the experiments were the number of authors and the number of writing samples per author. One of these parameters changed at a time to investigate the affect of a given parameter on the performance of the system.

We used a preprocessing method similar to [9] in all datasets. We replaced the numbers, username references, date, time, and website URLs with pre-defined meta tags. As we need as much information as we can get from the tweets, we did not convert the text into lowercase to keep the case information.

Table I shows the Twitter dataset [10] statistics for subsets of dataset used in the experiments with 100, 200, 500, and 1,000 authors (a). Similarly, Table II shows the dataset statistics for subsets of the dataset used in the experiments with 50, 100, 200, and 500 writing samples (w). Tables I and II have the same structure. The first column shows the number of authors (a) in Table I and the number of writing samples (w) in Table II. Then, the mean (μ) and standard deviation (σ) of the number of characters, words, and sentences for writing samples are shown. The last column of Tables I and II shows that the dictionary size ($|D|$) which is the number of unique words in the dataset.

In Table I, the average number of characters are within the range of 72 ± 2 for all authors. The character limit for Twitter is 140, the average number of characters are close to half the size of the limit. This trend is also observable in the average number of words and sentences. Please note that each author group is sampled without replacement, hence, they are disjoint sets.

On the other hand, the dictionary size grows when the number of authors increases which was expected since there are more writing samples for groups with more authors. The increase in dictionary size was less than the increase in the number of authors since it gets harder to find a unique word when the dictionary is larger.

TABLE I
DATASET STATISTICS: VARYING NUMBER OF AUTHORS

a	Characters		Words		Sentences		$\|D\|$
	μ	σ	μ	σ	μ	σ	
100	71.57	33.81	14.10	6.50	1.69	0.90	47493
200	73.15	34.04	14.29	6.49	1.71	0.96	81735
500	73.79	34.11	14.42	6.55	1.69	0.95	161742
1000	73.28	33.84	14.30	6.49	1.71	0.96	269774

In Table II, the average number of characters are within range of 73 ± 0.5 for all values of w. In addition, the dictionary size gets larger with more writing samples. Similar to the varying number of authors, an increase in writing samples results in a larger dictionary size.

TABLE II
DATASET STATISTICS: VARYING NUMBER OF WRITING SAMPLES

w	Characters		Words		Sentences		$\|D\|$
	μ	σ	μ	σ	μ	σ	
50	73.26	34.22	14.17	6.45	1.66	0.94	10857.50
100	73.04	34.20	14.15	6.45	1.65	0.92	18093.50
200	73.04	34.16	14.15	6.44	1.65	0.93	29941.50
500	72.87	34.11	14.12	6.43	1.65	0.92	56974.60

Our experimental settings are as follows.

A. Experiment I: Varying Number of Authors

We explored how our proposed architecture performs with an increasing number of authors. We performed a set of performance evaluations with a different number of authors while keeping the number of writing samples constant. The number of writing samples per author (w) was 200. The number of authors (a) was used in experiments was 100, 200, 500, and 1,000. In this experiment, we used the same sampling used in [9]. As expected, the reproduced results in this paper were similar to the reported results in [25] and [9]. We used 10-fold cross-validation on the experiments for each author group. Hence, we evaluated 40 train-test splits in total.

B. Experiment II: Varying Number of Writing Samples

In this experiment, we investigated the impact of a different number of writing samples for author identification. We performed a set of performance evaluations with a varying number of writing samples per author where the number of authors was held to 50. The number of writing samples per author used in experiments was 50, 100, 200, and 500. We sampled 10 different disjoint sets of groups for each writing sample size. We used 10-fold cross-validation on the experiments for each author group. Hence, we evaluated 400 experiments in total.

V. RESULTS

We compare our proposed method with the following baselines. The information about the algorithms that were used in the experiments can be found in the following list:

- K-Signatures [9]: Uses the features that include character and word N-grams. The features used by a single author at least K% of the documents are used. Also, a method called Flexible Patterns was utilized. Patterns in this method can match partially and get a score based on the closeness. Combination of K-Signatures and Flexible Patterns techniques shown better results in the paper. Hence, reported results are a combination of the techniques.
- LSTM$_{Char2}$: Long Term Short Term Memory (LSTM) networks widely used and known for the sequence to sequence tasks. There were also applications for Author Identification. We used character bigrams as an input to the network based on our preliminary findings on our development set.
- Rocha et al. [3]: Character 4-grams, word 1-5 n-grams, and POS 1-5 n-grams are extracted from the writing samples. Hapax legomena, the features are only seen once in the dataset, are removed to reduce the noise. Then, the extracted features are fed to PMSVM for the classification.
- Theóphilo et al. [4]: CNN architecture that uses character 4-grams as input. The layers used in the architecture were dynamic k-max-pooling [37] to capture the most important features in each layer and uses a technique called folding [37] that applies element-wise sum operation for every two rows. The same hyper-parameters that were used as in Theóphilo et al. [4]. The only difference in terms of hyper-parameters was the number of epochs. The number of epochs increased to 100 epochs instead of 10 epochs. This change was necessary for fairness since the algorithms compared against allowed to go as high as 100 epochs. The best performing model in the validation set was selected and evaluated on the test set.
- CNN$_{CharN}$ [25]: All character N-grams are randomly initialized and then updated in the training. The other parameters of the networks like filter weights are also updated with backpropagation. In this paper, we implement both CNN$_{Char1}$ and CNN$_{Char2}$ methods to compare with our proposed method.
- CNN$_{W2V}$ [26]: Pre-trained vectors in CNN$_{W2VStatic}$ are updated during the training. The pre-trained embeddings are trained on the Twitter dataset.
- CNN$_{FastText}$: All words are initialized with a pre-trained word vector from FastText. The pre-trained embeddings are trained on the Twitter dataset.

- CNN_{WC} [10]: Word embeddings initialized with pre-trained FastText embeddings and character bigrams are randomly initialized. Convolutional filters are shared with the embeddings. The embeddings and network parameters updated with backpropagation.

A. Results of Experiment I: Varying Number of Authors

Table III shows the results of approaches discussed in Section IV with varying number of authors. In Table III, the first column lists the algorithm used in the experiment, where the baseline methods are listed above the double lines and our proposed method is listed below the double lines. The rest of the columns list the accuracies of the algorithms with 100, 200, 500, and 1000 authors, respectively. The number in green marks the highest accuracy in each column.

The accuracies of the algorithms decrease when there is an increase in the number of authors for all algorithms. CNN_{WC} was the best performer among the baseline methods. $\text{CNN}_{FastText}$ outperformed the other baseline methods without using character-level information. $\text{CNN}_{FastText}$ was followed by CNN_{Char2} which uses Character Bigrams.

The proposed method, AARef, was the best performer for all authors sets in Experiment I. The accuracies of the AARef algorithms with 100, 200, 500, and 1000 authors are 57.18%, 55.31%, 47.99%, 42.83%, respectively. On average, AARef performed better than the state-of-the-art results with 100, 200, 500, and 1000 authors.

TABLE III
PERFORMANCES OF THE ALGORITHMS WITH VARYING NUMBER OF AUTHORS

Algorithms	Authors			
	100	200	500	1000
CNN_{Char1} [25]	49.24%	47.68%	41.37%	35.60%
CNN_{Char2} [25]	49.96%	48.84%	42.92%	37.55%
CNN_{W2V}	47.21%	45.52%	39.85%	34.73%
$\text{CNN}_{FastText}$	51.83%	50.25%	44.18%	38.74%
CNN_{WC} [10]	55.20%	53.14%	46.90%	41.28%
Rocha et al. [3]	43.99%	42.32%	36.63%	31.61%
Theóphilo et al. [4]	30.99%	31.44%	29.88%	27.76%
K-Signatures [9]	42.50%	41.10%	35.50%	30.30%
LSTM_{Char2}	33.80%	33.50%	29.80%	24.80%
AARef	**57.18%**	**55.31%**	**47.99%**	**42.83%**

B. Results of Experiment II: Varying Number of Writing Samples

Table IV shows the results of approaches discussed in Section IV with varying number of writing samples per author. In Table IV, the first column lists the algorithm used in the experiment, where the baseline methods are listed above the double lines and our proposed method is listed below the double lines. The rest of the columns list the accuracies of the algorithms with 50, 100, 200, and 500 writing samples per author, respectively. The number in green marks the highest accuracy in each column.

One can see that the accuracies increase as the number of writing samples increases for all the methods listed. Among the baseline methods, CNN_{WC} performed better than others. The performances of $\text{CNN}_{FastText}$, CNN_{Char1}, and CNN_{Char2} was very close. One can see that $\text{CNN}_{FastText}$ scaled better since $\text{CNN}_{FastText}$ was clearly better than CNN_{Char1}, and CNN_{Char2}.

The proposed method, AARef, was the best performer for all writing samples per author in Experiment II. The accuracies of the AARef algorithms with 50, 100, 200, and 500 writing samples per author are 56.10%, 63.51%, 70.05%, 76.04%, respectively. On average, AARef performed better than the state-of-the-art results with 50, 100, 200, and 500 writing samples per author. It is should be noted that, AARef had smaller footprint compared to Rocha et al. [3], where frequency of each character 4-grams, word 1-5 n-grams, and POS 1-5 n-grams were used as inputs.

TABLE IV
PERFORMANCES OF THE ALGORITHMS WITH VARYING NUMBER OF WRITING SAMPLES

Algorithms	Writing Samples			
	50	100	200	500
CNN_{Char1} [25]	51.40%	58.20%	64.07%	70.30%
CNN_{Char2} [25]	51.56%	58.25%	63.59%	69.80%
CNN_{W2V}	49.14%	56.68%	62.96%	69.70%
$\text{CNN}_{FastText}$	51.46%	59.14%	65.61%	72.46%
CNN_{WC} [10]	54.36%	62.17%	68.34%	74.50%
Rocha et al. [3]	42.88%	49.90%	57.43%	66.71%
Theóphilo et al. [4]	30.20%	38.32%	45.53%	56.00%
AARef	**56.10%**	**63.51%**	**70.05%**	**76.04%**

VI. CONCLUSION AND FUTURE WORK

We proposed a Multi-Channel CNN approach that take advantage of multi branch refinement convolution blocks. The architecture of proposed block includes different convolution branches with identity mapping. The skipped connection in the identity mapping block helps the model capture salient information and passes the information on to the rest of network. We compared the performance of AARef with state-of-the-art CNNs that uses a Multi-Channel architecture. The results of the experiments show that the AARef performs better than the state-of-the-art CNNs on the Twitter dataset.

In the future, it would be interesting to see how network will perform with the conceptualized embeddings.

REFERENCES

[1] E. Stamatatos, "A survey of modern authorship attribution methods," *Journal of the American Society for information Science and Technology*, vol. 60, no. 3, pp. 538–556, 2009.

[2] T. Neal, K. Sundararajan, A. Fatima, Y. Yan, Y. Xiang, and D. Woodard, "Surveying stylometry techniques and applications," *ACM Computing Surveys (CSUR)*, vol. 50, no. 6, p. 86, 2018.

[3] A. Rocha, W. J. Scheirer, C. W. Forstall, T. Cavalcante, A. Theophilo, B. Shen, A. R. Carvalho, and E. Stamatatos, "Authorship attribution for social media forensics," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 1, pp. 5–33, 2016.

[4] A. Theóphilo, L. A. M. Pereira, and A. Rocha, "A needle in a haystack? harnessing onomatopoeia and user-specific stylometrics for authorship attribution of micro-messages," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 2692–2696.

[5] S. Nizamani and N. Memon, "Ceai: Ccm-based email authorship identification model," *Egyptian Informatics Journal*, vol. 14, no. 3, pp. 239 – 249, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S111086651300039X

[6] S. Phani, S. Lahiri, and A. Biswas, "A machine learning approach for authorship attribution for bengali blogs," in *2016 International Conference on Asian Language Processing (IALP)*, 2016, pp. 271–274.

[7] R. Overdorf and R. Greenstadt, "Blogs, twitter feeds, and reddit comments: Cross-domain authorship attribution," *Proceedings on Privacy Enhancing Technologies*, vol. 2016, no. 3, pp. 155–171, 2016.

[8] K. Luyckx, *Scalability issues in authorship attribution*. ASP/VUBPRESS/UPA, 2011.

[9] R. Schwartz, O. Tsur, A. Rappoport, and M. Koppel, "Authorship attribution of micro-messages," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 1880–1891.

[10] S. Aykent and G. Dozier, "Author identification of micro-messages via multi-channel convolutional neural networks," *IEEE Transactions on Systems, Man, and Cybernetics*, 2020.

[11] K. Luyckx and W. Daelemans, "The effect of author set size and data size in authorship attribution," *Literary and linguistic Computing*, vol. 26, no. 1, pp. 35–55, 2011.

[12] V. Kešelj, F. Peng, N. Cercone, and C. Thomas, "N-gram-based author profiles for authorship attribution," in *Proceedings of the conference pacific association for computational linguistics, PACLING*, vol. 3. sn, 2003, pp. 255–264.

[13] M. Koppel, J. Schler, and S. Argamon, "Authorship attribution in the wild," *Lang. Resour. Eval.*, vol. 45, no. 1, pp. 83–94, Mar. 2011. [Online]. Available: http://dx.doi.org/10.1007/s10579-009-9111-2

[14] W. J. Teahan and D. J. Harper, "Using compression-based language models for text categorization," in *Language modeling for information retrieval*. Springer, 2003, pp. 141–165.

[15] E. Stamatatos, "Author identification using imbalanced and limited training texts," in *18th International Workshop on Database and Expert Systems Applications (DEXA 2007)*. IEEE, 2007, pp. 237–241.

[16] D. Benedetto, E. Caglioti, and V. Loreto, "Language trees and zipping," *Physical Review Letters*, vol. 88, no. 4, p. 048702, 2002.

[17] T. Joachims, "Making large-scale svm learning practical," Technical Report, Tech. Rep., 1998.

[18] C. Apte, F. Damerau, S. Weiss *et al.*, *Text mining with decision rules and decision trees*. Citeseer, 1998.

[19] H. T. Ng, W. B. Goh, and K. L. Low, "Feature selection, perceptron learning, and a usability case study for text categorization," in *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, 1997, pp. 67–73.

[20] J. Diederich, J. Kindermann, E. Leopold, and G. Paass, "Authorship attribution with support vector machines," *Applied intelligence*, vol. 19, no. 1-2, pp. 109–123, 2003.

[21] I. Khomytska and V. Teslyuk, "Authorship attribution by differentiation of phonostatistical structures of styles," in *2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT)*, vol. 2. IEEE, 2018, pp. 5–8.

[22] U. Sapkota, S. Bethard, M. Montes, and T. Solorio, "Not all character n-grams are created equal: A study in authorship attribution," in *Proceedings of the 2015 conference of the North American chapter of the association for computational linguistics: Human language technologies*, 2015, pp. 93–102.

[23] A. Narayanan, H. Paskov, N. Z. Gong, J. Bethencourt, E. Stefanov, E. C. R. Shin, and D. Song, "On the feasibility of internet-scale author identification," in *2012 IEEE Symposium on Security and Privacy*, May 2012, pp. 300–314.

[24] J. Wu, "Power mean svm for large scale visual classification," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 2344–2351.

[25] P. Shrestha, S. Sierra, F. González, M. Montes, P. Rosso, and T. Solorio, "Convolutional neural networks for authorship attribution of short texts," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 669–674. [Online]. Available: https://www.aclweb.org/anthology/E17-2106

[26] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1746–1751. [Online]. Available: https://www.aclweb.org/anthology/D14-1181

[27] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of machine learning research*, vol. 12, no. Aug, pp. 2493–2537, 2011.

[28] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[29] D. Bagnall, "Author identification using multi-headed recurrent neural networks," *arXiv preprint arXiv:1506.04891*, 2015.

[30] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[31] Y. Hoshen, R. J. Weiss, and K. W. Wilson, "Speech acoustic modeling from raw multichannel waveforms," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4624–4628.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[33] O. Owoputi, B. O'Connor, C. Dyer, K. Gimpel, N. Schneider, and N. A. Smith, "Improved part-of-speech tagging for online conversational text with word clusters," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, Georgia: Association for Computational Linguistics, Jun. 2013, pp. 380–390. [Online]. Available: https://www.aclweb.org/anthology/N13-1039

[34] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 971–980. [Online]. Available: http://papers.nips.cc/paper/6698-self-normalizing-neural-networks.pdf

[35] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, no. 1, 2013, p. 3.

[36] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of machine learning research*, vol. 13, no. Feb, pp. 281–305, 2012.

[37] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, Jun. 2014, pp. 655–665. [Online]. Available: https://www.aclweb.org/anthology/P14-1062